SANDIA REPORT

SAND2003-3612
Unlimited Release
Printed October 2003

Xyce Release and Distribution Management

Version 1.2

C. Mike Williamson and Scott A. Hutchinson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-3612
Unlimited Release
Printed October 2003-10-13

Xyce™ Release and Distribution Management Version 1.2

C. Michael Williamson
Software and Information Engineering

Scott A. Hutchinson
Computational Sciences

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0316

Abstract

This document presents a high-level description of the **Xyce**™ Parallel Electronic Simulator Release and Distribution Management Process. The purpose of this process is to standardize the manner in which all **Xyce** software products progress toward release and how releases are made available to customers. Rigorous Release Management will assure that **Xyce** releases are created in such a way that the elements comprising the release are traceable and the release itself is reproducible. Distribution Management describes what is to be done with a **Xyce** release that is eligible for distribution.

Acknowledgements

The following people have contributed to defining and documenting the **Xyce** Release and Distribution Management process described in this document:

- Molly Ellis of the Information Systems Development Department.
- Ann Hodges, Shelly Eaton, Mar McCormack, Gary Froehlich and Mike Williamson of the Software and Information Engineering Department.

Trademarks

Xyce™ Parallel Electronic Simulator and **Xyce**™ are trademarks of Sandia Corporation.

Contacts

Email	xyce-support@sandia.gov
World Wide Web	http://www.cs.sandia.gov/Xyce

Release and Distribution Management Process Change History

Version	Author	Modifications	Date
1.0	Mike Williamson	Initial Release	5/1/2001
1.1	Scott Hutchinson	Modifications for **Xyce**	3/7/2002
1.2	Scott Hutchinson	Modifications for **Xyce** to support machine-based distributions	7/31/2003

Contents

Preface

This document presents a high-level description of the **Xyce** Parallel Electronic Simulator Release and Distribution Management Process. The purpose of this process is to standardize the manner in which all **Xyce** software products progress toward release and how releases are made available to customers. Rigorous Release Management will assure that **Xyce** releases are created in such a way that the elements comprising the release are traceable and the release itself is reproducible. Distribution Management describes what is to be done with a **Xyce** release that is eligible for distribution.

1. Introduction

1.1 Document Purpose

The purpose of this document is to provide a high-level description of the **Xyce** Parallel Electornic Simulator Release and Distribution Management (RDM) process.

This document describes the process elements of RDM, but it does not describe their specific implementation. No specific tool is recommended or discussed in detail. There are several implementations that would successfully carry out the process described in this document. Therefore, even if the implementation or the toolset changes in the future, this process remains stable.

1.2 Scope

The **Xyce** Release and Distribution Management (RDM) process described in this document is required for each **Xyce** product. The main sections of this document focus on the tasks and activities that are required for RDM. Templates and project specific extensions are included in the appendices of this document.

This RDM process applies to all of the work products of application engineering, software development, and verification. A subset of these work products and artifacts are included in the product release that is distributed to customers. The specific subset of work products and artifacts included in the product release is dependent upon the product being released, the intended use, and the customer. These dependencies should be captured in the Requirements Management Process.

RDM consists of two processes: the Release Management Process (Section 3) and the Distribution Management Process (Section 4). The Release Management Process identifies the activities that must be addressed for a product release to become eligible for distribution. The Distribution Process describes the activities that must be addressed when distributing a product release to customers.

Release Management Process

The Release Management Process identifies the objectives, goals and activities that need to be addressed for a product release to be eligible for distribution. The output of the Release Management Process is a product release that is eligible for distribution.

The Release Management Process relates to activities that occur internally to the **Xyce** project. It includes activities that must begin when a new software release is initially foreseen, as well as activities up to and including the point in time when the new product release is eligible for distribution to customers.

The outputs of the Release Process include:

- **Xyce product release(s),** comprising features to meet customer requirements, enhancement requests and potentially fixes for specific defects.
- **Installation notes,** describing installation activities for the release.
- **Release notes,** describing new features, changes to existing features, defect fixes, deficiencies, known defects and limitations.

The Release Management Process is described in Section 3. The complete set of artifacts that comprise a product release are not defined in this process. Refer to Section 6 for a list of the interfaces between this document and other **Xyce** processes.

Distribution Management Process

The Distribution Management Process describes the objectives, goals and activities that need to be addressed once a product release is eligible for distribution to customers. The output of the Release Management Process—a product release—is an input to the Distribution Management Process. The outputs of the Distribution Management Process are data items that are used to track and manage product distributions (i.e. track and manage which customers have which release artifacts).

The Distribution Management Process describes the needs for a product becomes eligible for distribution, when the product is distributed, and when a product is withdrawn. The Distribution Process is covered in Section 4.

Roles

Roles are functional divisions of the process responsibilities and they are defined based on the activities that comprise RDM. Roles are covered in Section 5.

1.3 Goals

Since 2001, Sandia National Laboratories (Sandia) has been developing a DOE-ASCI funded parallel electric circuit simulation code named **Xyce**™. The **Xyce**™ Parallel Electronic Simulator has been written to support, in a rigorous manner, the simulation needs of the Sandia National Laboratories electrical designers. The code has been developed using an object-oriented design and modern coding-practices that ensure that the **Xyce** will be maintainable and extensible far into the future.

Developing and modifying the **Xyce** products in time frames responsive to the customer can be a significant effort, so the procedures to manage the items that make up the **Xyce** products need to be as clear and non-intrusive as possible so that efficient response to the customer can be maintained.

The primary goal is to document the Release and Distribution Process for the managers and developers of **Xyce**.

Thus, the main goals of the RDM process are to:

- Ensure that all **Xyce** releases contain the proper artifacts;
- Ensure that all **Xyce** releases are reproducible;
- Ensure that all phases of the **Xyce** release life cycle are adequately tracked and documented;
- Define a process that is adequate for current **Xyce** needs and which can be enhanced with additional procedures at a later date if desired;
- Ensure that distributions of all **Xyce** product releases can be tracked; and
- Allow prior releases of **Xyce** products to be withdrawn if needed.

2. Abbreviations and Definitions

artifact

A deliverable or work product that is the output of some phase of the software development life cycle. A *configuration-controlled artifact* is an artifact that is stored in a corporate repository (library), and changes to it are controlled.

authorized maintiner

The person responsible for installing and maintaining the **Xyce** application on the platform or specific environment. This person is also a customer.

baseline

(verb) To capture a snapshot of a controlled item (or group of controlled items) at a reference point within the item's development life cycle. (noun) A reference point in the development of a controlled item or the snapshot of the controlled item captured at a reference point within the item's development life cycle.

change request

Documentation (formal or informal) of enhancements, modifications, or bug fixes being succested for the system. The change request must be approved prior to the work specified in the change request beginning. Examples include formal change control documents, issues (bugs), and enhancement requests.

check in (commit)

To put the initial or a new revision of an element into a version control system.

check out

To extract a revision of an element from a version control system.

components

Tightly coupled/interdependent sets of modules that provide functionality. Capabilities trace to components.

customer

Any person who wants to use the software application, report a problem, request an enhancement, or request a specific version of the application.

distribute

To make a product release available to a customer. Depending on the customer's needs, the customer may pull the product distribution or **Xyce** may push the distribution to the customer.

distribution management

Establishing, maintaining, and tracking procedures, roles, and responsibilities for distribution of products to customers.

element

Lowest level (atomic) item that is subject to version control.

element version

The revision of an element. Used when referring to a specific revision of an element; such as, "What version of the file is in the release?"

emergency patch

A patch that has to be implemented and released as quickly as possible because critical operations, decisions, or results are impacted. Contrast to *immediate patch*.

full release

A release that is self contained (i.e. the release may be installed and used without access to any previous release).

freeze

To prevent additional changes to a specific version of a version controlled item.

immediate patch

A patch that needs to be implemented before the next scheduled primary release.
, but the updates are not critical. Contrast to *emergency patch*.

label

An identifying marker that can be associated with specific versions of version controlled elements. Multiple elements can be assigned the same label as a means of grouping the elements.

locked

An indicator that a specific reversion of a version controlled element may not be modified.

major release

A scheduled or planned release of a product that comes out when there are extensive new product features, when there is a significant redesign of the product or when customers of the product are required to make significant changes in how they use the product or in support elements of the product such as the version of a supporting commercial application or third party software. See definitions for *minor release* and *patch release*, Figure 1.

minor release

A scheduled or planned release of a product that comes out when a product feature has been added or significantly modified from its original documented behavior. A minor release usually does not imply significant redesign of the product although there may be redesign of some of its components. A minor release also should not require the customer to make significant changes in how they use or support the product. See *major release* and *patch release*. See Figure 1.

module

Smallest coherent unit of a product. Requirements trace to modules. Modules are composed of one to many version controlled elements.

partial release

A release that contains some subset of a products elements and/or components. Partial releases must be installed over, or in conjunction with, a previous full release. See *full release*.

patch

An *as-needed* update to one or more of the elements that comprise a product for the express purpose of fixing critical or *function-impacting* defects. See Figure 1.

patch release

A product release that is generated due to a patch. See *release* and *primary release*.

primary release

A scheduled and planned release of the product which is either a major release or a minor release. A release is categorized as either a *primary release* or a *patch release*. See *release* and *patch release*.

product

Items offered for use by licensed customers. **Xyce** products will be the **Xyce** Framework, ASCI Applications that utilize the **Xyce** Framework, third-party libraries, and **Xyce** Tools.

product release

(formal) A captured occurrence of a product. Release is used in referring to the identification of a product; such as, "What is the release of the product?"

promotion

Moving from one release development life cycle activity to a higher release development life cycle activity.

resource distribution

A *distribution* of the application provided as a resource on specific simulation platforms, for specific environments, for use by *authorized* persons with accounts on the platform. These distribtutions are maintained by an authorized *maintainer*.

release

An integrated set of one to many products that will, when ready, be made available for distribution to customers. See *full release*, *partial release*, *primary release*, and *patch release*.

release management

Establishing, maintaining, and tracking procedures, release development lifecycle activity, roles, and responsibilities for releasing product to customers.

release number

The alpha-numeric identifier given to a specific product release.

role

A functional division of responsibilities.

subsystem

An architecturally motivated organization of components.

tasks

The major activities that are performed in the release process.

version

One of a sequence of copies of an element, each incorporating modifications. See *product release* and *element version*.

version control

Identifying, maintaining, and tracking versions of the components of a product and versions of the product itself.

version-controlled file

A file that has been placed under the version control system.

version number

A numeric identifier assigned to a specific occurrence of an element or product. See *product release* and *element version*.

withdraw

To make a specific product release ineligible for distribution. A product release that has been withdrawn is no longer supported.

3. Release Process

3.1 Objective

The objective of the Release Process is to define the activities in the Release Process with adequate detail to facilitate the process implementation.

3.2 Goals

The goals of the Release Process are to:

- Define Release Process activities;
- Identify roles that are responsible for the activities.

3.3 Process

A simple definition of a release is a version of a product that will, when ready, be made eligible for distribution.

This definition of a release is very general and has broad application. For example, the release might be a single element, a group of elements, an application, or a suite of applications.

Xyce products that are subject to this process include the **Xyce** Parallel Electronic Simulator as well as any associated third-party software (TPS). **Xyce** release customers are internal Sandia analysts and designers as well as non-Sandians associated with the ASCI program. This may be expanded in the future to include external customers.

Prior to defining the activities in the Release Process, it is necessary to explain release types , release numbering and baselining.

Release Types

Software products are frequently refined, enhanced, and fixed. Consequently, new versions of products are apt to become available for release. Releases may be pre-planned, where the features are outlined in an overall strategy for the product, or the releases may be extemporaneous to fix issues in a current release. Full releases will contain all product components, and partial releases will contain some subset of product elements and components.

Figure 1 shows the relationship between types of releases.

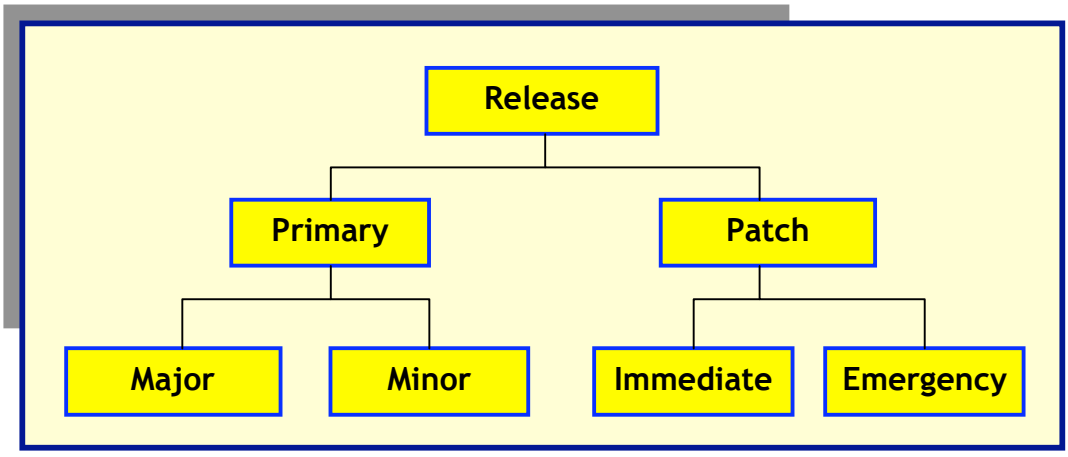

Figure 1. Release Hierarchy

Primary Release

A release that is pre-planned is called a *primary release*. A primary release can be either a *major release* where the product has had significant changes made to it or a *minor release* where the changes are more incremental in nature.

All primary releases need to be made available as full releases (they contain all components required to install and use the product). A primary release must include the features and fixes of all previous primary and patch releases, unless the feature is being discontinued.

Xyce primary releases will occur periodically. The specific release schedule for primary releases is outside the scope of this document.

Patch Release

An extemporaneous release that is implemented to fix issues in an existing release is known as a *patch release*. The two types of patch releases are *immediate patch* and *emergency patch*. Patch releases may be full or partial releases (depending on project need) but will generally be partial releases.

An *immediate patch* is indicated when the patch release must be eligible for distribution prior to the forecast distribution date of the next planned release, and the need for the patch release is not critical enough to justify an emergency patch. All normal Release Process steps are followed for an immediate patch.

An *emergency patch* is indicated when the patch release must be eligible for distribution as quickly as possible. When generating an emergency patch, some of the activities normally required for a product release may be deferred. Deferral of process activities involves risk management decisions and must be preceded by appropriate approvals. All deferred activities must be completed retroactively after the release. Emergency patches should be rare as they are only generated in critical situations.

Release Numbering

Release numbering is the way a release is named and labeled so that it can be uniquely identified and referenced. The release number format for **Xyce** is the following:

(baseline) MajorRelease.MinorRelease.Patch.ModificationType

Table 1 describes the terms in the release number format. Refer to Table 5 for information on the release development life cycle activities.

Term	Description
Baseline	Indicates the intermediate release development life cycle activity (S for STABLE, Q for QA). This term is not used for PROD releases.
MajorRelease	A numeric term that is required for all releases.
MinorRelease	A numeric term that is required for all releases.
Patch	A numeric term that is optional if the patch number is 0.
Modification	A numeric term that is used for baseline identification and is always suppressed in PROD release labels.
Type	Alpha-numeric term that is used to identify sub-varieties of the release. This term is intentionally flexible as release types may surface at any point during the release life cycle and packaging.

Table 1. Release Numbering Terms

Table 2 provides a few examples of baseline and release numbers for a fictitious **Xyce** product called Jasmine.

Description	Sample Release Number
Stable release of **Xyce** 1.0	(S) Xyce 1.0
QA release of **Xyce** 1.0	(Q) Xyce 1.0
Production release of version 1.0	Xyce 1.0
First production patch release	Xyce 1.0.1
First production minor release	Xyce 1.1
Version 1.1 packaged on CD for run-time-only licensed customers	Xyce 1.1 RTO-CD

Table 2. Release Number Examples

Baselining

Baselining is defined as capturing a snapshot of a controlled element (or controlled elements) at some reference point. Baselining provides a mechanism for logical groupings of elements, such as the elements that compose a product release. Baseline creation can be triggered by events in the release development life cycle as well as date or milestone events.

Baselines are comprised of a single revision of at least one version controlled element. A single revision of an element may be a member of multiple baselines. However, only one revision of each element may be in any specific baseline. In other words, if a revision of an element is denoted as being in a specific baseline, no other revision of that element can be in the same baseline.

What to Baseline

All elements that comprise **Xyce** product releases will be baselined during the appropriate release activity. There is a hierarchy of relationships that goes from the element level to the product:

- Element - the lowest level (atomic level) object that is subject to version control. Examples of elements include: source code files, build dependency files, build scripts and important outputs of release activities (artifacts).

- Module - composed of one to many elements. Modules are the smallest coherent unit of a product.

- Component - Tightly coupled/interdependent sets of one to many modules that provide functionality. Capabilities trace to components.

- Subsystem - An architecturally motivated organization of one to many components.

- Product - Items offered for use by licensed customers. **Xyce** products will be the **Xyce** Parallel Electronic Simulator and associated third-party libraries. A product is comprised of from one to many subsystems.

Not all version controlled elements will be in a baseline. For example, when an element is initially placed under version control, it is not yet in a baseline. When the element is identified as being part of an intended release, the element will be baselined into a STABLE baseline.

There are two general categories of baselines: *release* and *reference*.

Release Baselines

A release baseline is created when a **Xyce** product, or a component of a **Xyce** product, completes the STABLE, QA, or PROD release development life cycle activities.

Reference Baseline

A reference baseline is created any time a traceable and reproducible reference point is desired. Possible examples include representing the product state as of a specific date or milestone.

Modifying a Baseline

With only one exception, baselines are never modified. The single exception relates to release baselines. Release baselines may be modified in that they may have elements that represent artifacts of the current activity added to the baseline (e.g. test results, installation instructions). Once a product release has been moved to the

next release development life cycle activity, all previous release development life cycle activity release baselines are permanently frozen.

Baseline Abandonment

There are times when events require that a release baseline be abandoned. Abandoning a baseline implies that the product release is being returned to the previous release life cycle activity. Reference baselines are never abandoned.

For example, if a release baseline fails to pass all acceptance criteria, then that release baseline is not accepted. Notification that the release baseline has failed is passed back to the previous development life cycle activity along with results for the criteria that were failed. The specific release baseline is said to be abandoned. The release baseline is retained for historical record.

When a release baseline fails acceptance criteria and is abandoned, element fixes occur in the previous release development lifecycle activity. A new release baseline is created that includes the fixes. The baseline *modification* term is incremented so that every baseline is uniquely identified.

Baseline Identification

The standard that will be used for identifying baselines should be flexible enough to uniquely identify all varieties of product and reference baselines. The **Xyce** baseline identification format is:

ID Type release-number

Table 3 describes the terms in the baseline identification.

Term	Description
ID	An alpha-numeric string that identifies the product for a release baseline, and is a freeform identifier for reference baselines. Legitimate values for release baselines will be Xyce and <TPS NAME>.
Type	A literal string that is used to indicate the type of baseline. - *Rel* for release baselines - *Ref* for reference baselines
Release-number or text	The number described in Table 1. This term is used for release baselines only (i.e. it is not used for reference baselines).

Table 3. Baseline Identification Terms

Since the baseline identification must be flexible enough to accommodate both product and reference baseline labeling, a few examples may be useful. Table 4 contains baseline identification examples.

Description	Baseline Identifier
Second DEVEL iteration for release baseline of **Xyce** version 1.0.	(D) Xyce Rel 1.0.0.1
First Stable release baseline of **Xyce** version 1.0.	(S) Xyce Rel 1.0.0.0
Fourth QA release baseline of **Xyce** version 1.0.	(Q) Xyce Rel 1.0.0.4
Production release baseline of version 1.0.	Xyce Rel 1.0.0
Date specified reference baseline (DOE literal is intended to represent some request from DOE to checkpoint progress at a point in time).	DOE Ref Jan 4, 2001
Reference baseline for **Xyce** product that is not based on the release life cycle. The text "for media test" is the term "release-number or text."	Xyce Ref for media test

Table 4. Baseline Identification Examples

Release Activities

(Scott – something is wrong with the figure reference)The Release Process for **Xyce** consists of activities and the roles that are primarily responsible for those

activities. Each activity has entry and exit criteria. Activities are performed in a specific order, with a limited number of paths to completion of the release request.

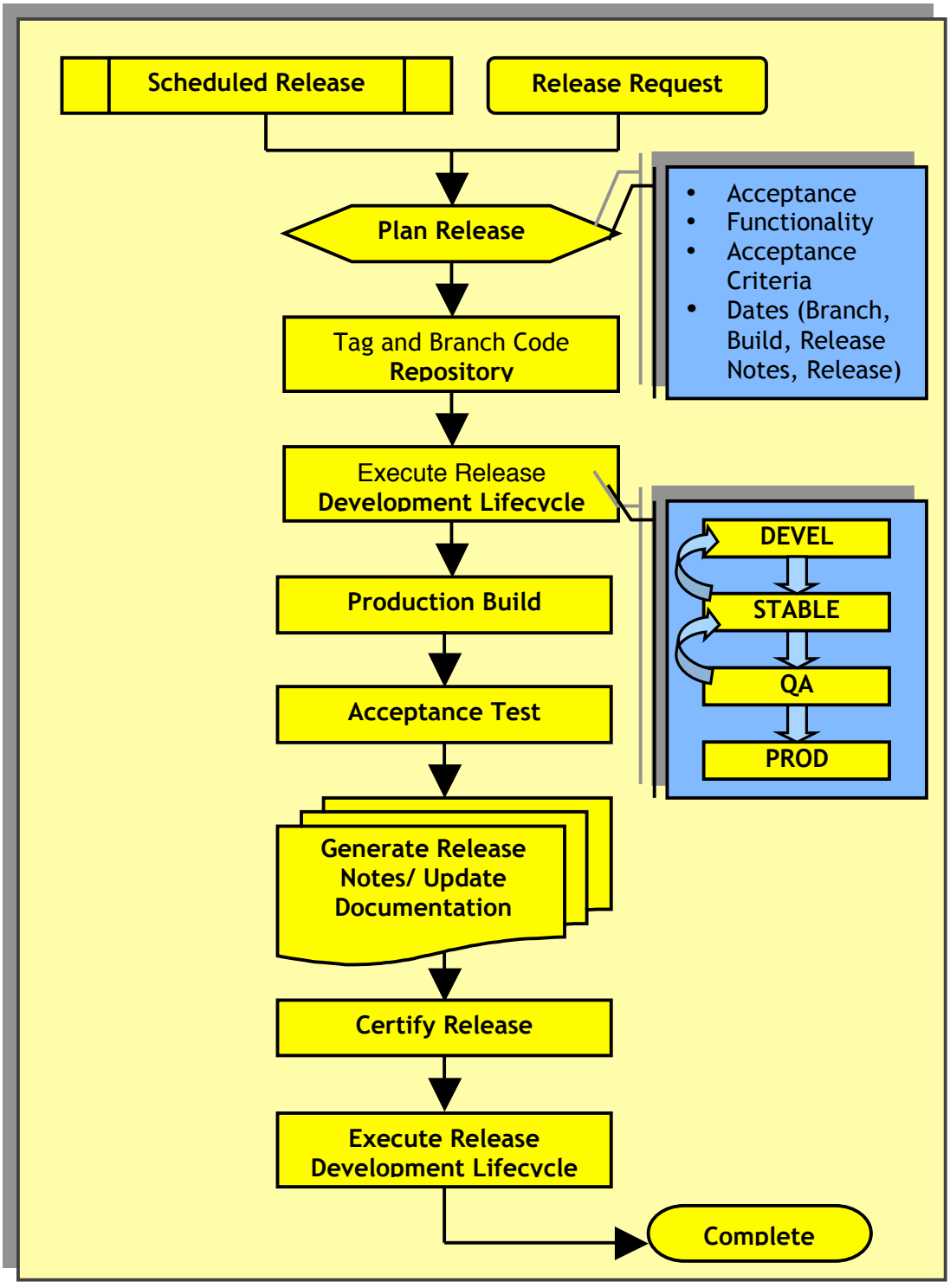

Figure 2. Release Process

Figure 2 illustrates the Release Process and shows the paths that a release request can take from submission to closeout.

Products can be in multiple release cycles at the same time. For example, if a product has a release that is in the Release Development Lifecycle PROD activity and

new development has been started since the release, the product will also be in the Release Development Lifecycle DEVEL, STABLE, or QA activities. If the product is also being patched then there will be another patch release in one of the same activities. This represents three different releases (current release, next primary release, patch release) for the product.

(Scott – there is an error here and on the previous page. MW ->)While **Error! Reference source not found.** provides a good visual overview of the Release Process flow, it is not detailed enough to guide implementation. Table 5 gives more detail for the activities in the Release Process and identifies the role that is responsible for each activity. Additional information is provided for the four execute activities (DEVEL, STABLE, QA, PROD) that follow. Roles are defined in Section 5.

Activity	Description	Role Responsible
Submit	Submission of the Product Release Request begins the Release Process. The release request must contain specific enough information to allow the PTL to review and scope the request.	PTL or CCB
Review	Review includes scoping and requires careful study of the request, requirements, and other release requests. The request may be: ▪ Returned to the originator for more information; ▪ Combined with other requests; or ▪ Forwarded for approval.	PTL
Approve	Approval is required before the request moves into the activities that consume more resources. The request may be: ▪ Approved; ▪ Returned for more information; ▪ Deferred; or ▪ Rejected.	CCB
Plan	The planning activity includes: ▪ Resource planning and allocation; ▪ Generation of schedules; and ▪ Milestone identification.	PTL
Execute	The execute activity encompasses all activities of the Release Development Lifecycle. The activities in the Release Development Lifecycle are: ▪ DEVEL; ▪ STABLE; ▪ QA; and ▪ PROD More detail is provided for the activities in the Release Development Lifecycle below.	DEV & QA
Certify	Certification indicates that this is an appropriate time for the release to be made eligible for distribution.	RL
Notify	Interested parties are notified that the release is "eligible for distribution". One key interested party that is always notified is the Distribution Lead (DL)	RL

Close request	Process improvement tasks are performed. The release request is administratively closed.	RL

Table 5. Release Process Activities

Execute: DEVEL

The first activity executed in the Release Development Lifecycle is the DEVEL activity. During this activity releases are in said to be "in development". Releases in DEVEL are not subject to mandatory version control (although version control is recommended), or mandatory baselines. Work is done to add, enhance, fix, or remove functionality.

Releases may be returned to this activity from the Release Development Lifecycle STABLE activity.

All work done during this activity must be directly linked to some form of a change requests (new reqreuiement, bug reports, …). Any work that exceeds the scope of existing change requests mandates that new or updated change request(s) be generated.

Execute: STABLE

The second activity in the Release Development Lifecycle is the STABLE activity. Releases that have reached this activity are ready for element and component level testing. This is the first Release Development Cycle activity with a mandatory baseline, and from this point on all Release Development Lifecycle activities require a baseline.

Releases that fail to meet acceptance criteria are returned to the DEVEL activity with error descriptions and/or change requests. Similarly, releases may be returned to the STABLE activity from the Release Development Lifecycle QA activity.

All work done during this activity must be directly linked to change requests. Any work that exceeds the scope of existing change requests mandates that new or updated change request(s) be generated.

Execute: QA

The third activity in the Release Development Lifecycle is the QA activity. During this activity releases are built in an environment as close to the target platform as possible. All acceptance tests are executed.

If the release fails to pass any test criteria it is returned to the Release Development Lifecycle STABLE activity along with error descriptions and change requests.

Execute: PROD

The final activity in the Release Development Lifecycle is the PROD activity. The primary reason for this activity is to indicate that the release is ready to be certified. Various tasks may be completed during this activity including:

- Release notes;
- Installation documentation; and
- Planning for release packaging.

Product Release Progression

Products go through a progression of releases as time goes on. Major releases are succeeded by minor releases. Major and Minor releases are "patched". Figure 3 shows a typical product release progression.

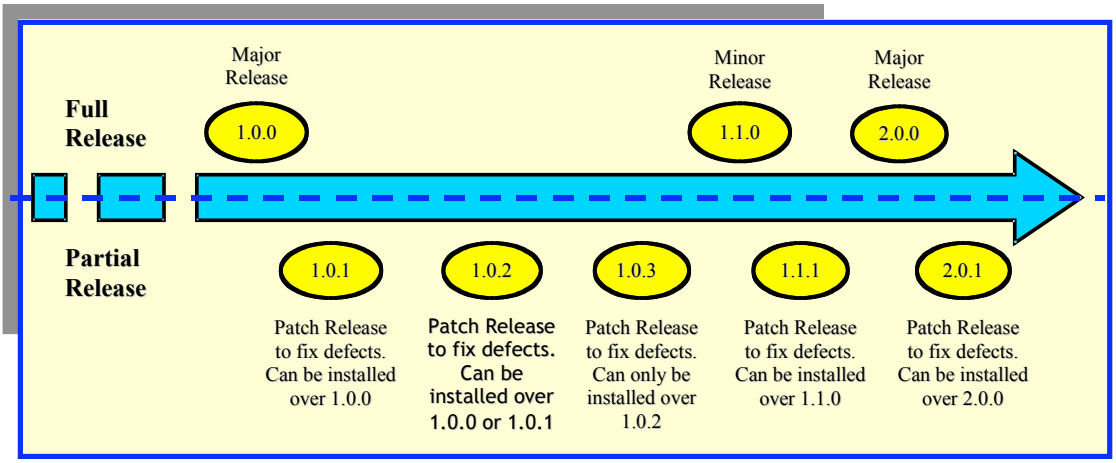

Figure 3. Example of a typical product release progression

4. Distribution Management Process

4.1 Objective

The objective of the Distribution Process is to define the activities in the distribution management process with adequate detail to facilitate the process implementation.

4.2 Goals

The goals of the Distribution Process are to:

- Outline activities, documentation and roles associated with providing released versions of the product(s) to the customers in a documented and controlled manner.

- Define high-level Distribution Process activities and the roles that are responsible for those activities;

- Establish documentation requirements for distribution management.

4.3 Process

Scope

This distribution management process applies to all **Xyce**-related product releases.

General

Once a release has successfully completed the Release Management certification activity it becomes eligible for distribution. Releases that are eligible for distribution may be moved to an electronic distribution system, packaged for distribution, shipped to authorized customers, or installed as a resource on specific simulation platforms by authorized maintainers (to be accessed by customers).

Distribution management deals with the packaging, shipping, tracking and notifications associated with releases. Distribution management does not allow for any changes to be made to the underlying release as part of the Distribution Process.

For the period of time that a release is to be available to customers the release is said to be "in distribution". If, for any reason, it is determined that a specific release should no longer be available for customers that specific release is said to be withdrawn. Withdrawn releases must not be distributed to any customer.

4.4 Requirements

The requirements for distribution management are as follows:

- All **Xyce**-related product distributions shall be controlled under this **Xyce** Distribution Process. As part of the release notes (and preferably as part of a product license) release customers should be advised that they are not allowed to redistribute any part of any **Xyce**-related product.

- Only those products that have successfully completed the certification activity of the Release Process are eligible for distribution.

- All requests for distributions and all actual distributions will be tracked. The minimum information required for distribution tracking includes customer name, customer location, **Xyce** product(s), release number, distribution date, media type, and target platform.

- Any release may be withdrawn from distribution. For example, an old release might be retired when a new release becomes eligible for distribution.

- Distribution must take place in such a way that only authorized customers receive the release.

4.5 Distribution Packaging

Prior to being distributed releases are packaged. Distribution packaging consists of:

- Identification of the correct product elements;
- Selection of media type based on customer requirements;
- Selection of a package format; and
- Packaging.

The key requirements for packaging are:

- All release packages will be properly labeled (physically and/or electronically) with the proper release identifier;
- The contents of all release packages will be properly documented; and
- Other than documentation activities (e.g. Release Notes), releases are never created or modified during the Distribution Process.

Media Type

The media type identifies the physical storage mechanism used by the release. Examples of media type include: CD, tape, and distribution server.

Package Type

Package type identifies a storage format. The three package types are multiple files, archive files, and mixed files.

A multiple file package contains all of the elements required to install the release with each file independently stored on the media.

An archive file is a single file that contains multiple files. Examples of archive files are tar files, ZIP files, and self contained installation files.

Mixed file packages will contain some combination of multiple files and archive files.

4.6 Distributing Releases

Once a release is eligible for distribution, customer requests for the release may be considered. Figure 4 illustrates the Distribution Process and the role(s) responsible for the activity. Roles are defined in Section 5.

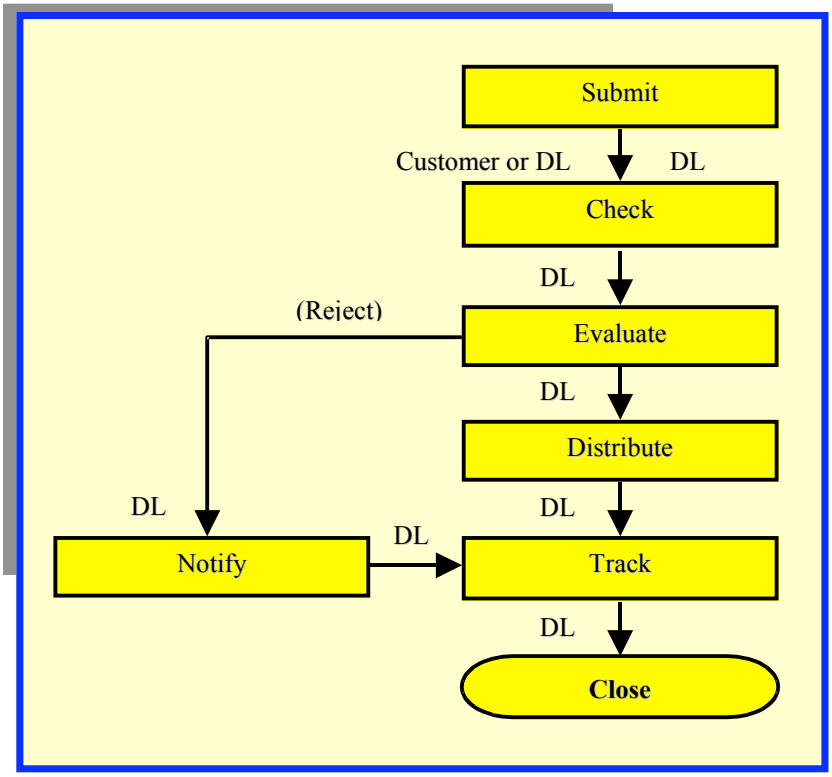

Figure 4. Distribution process

Figure 4 shows the process flow and the role(s) responsible for each activity. Additional details of each activity, the role with primary responsibility for the activity, and the transitions between activities are covered in Table 6.

Activity	Description	Role Responsible
Submit	The request must include the customer name, product release number, media type, and target system. The request may be automatically generated from customer license lists (standing orders), generated by an electronic customer interface, or generated as a result of verbal or textual communications with the customer.	Customer or DL
Check	All distribution requests are checked for completeness. The request may be returned to the submitter or transitioned to the next activity.	DL
Evaluate	Distributions may only be made to customers who are licensed and authorized to receive the specific release requested. In addition, only requests for releases that are "eligible for distribution" may be considered. Requests may be rejected or transitioned to the next activity.	DL
Distribute	There are numerous ways that the actual distribution may occur. While the method of distribution and the specific packaging may vary, in no case shall the underlying release be modified. See Section 4.6.1 and Section 4.6.2 for additional details.	DL
Track	All distribution requests are tracked. The minimum data that is retained is customer name, request date and request status.	DL
Close	Perform process improvement and administratively close the distribution request.	DL

Table 6. Distribution Process Detail

Distribution Mechanisms

Examples of distribution mechanisms include;

- Releases may be automatically packaged, copied to selected media, and shipped to specific customers;
- Releases may be packaged and electronically distributed (pushed) to specific customers; and
- Releases may be placed on a distribution server and made available for customers to "pull" at their convenience. In all cases, the distribution server software must authenticate customers prior to distribution.

4.6.2 Distribution Logging

All distributions will be logged. The minimum information that will be logged is:

- Customer name;
- Authentication source;
- Specific release (product and version);
- Distribution media;
- Distribution packaging;
- Shipment mechanism; and
- Distribution date.

4.7 Withdrawing Releases From Distribution

At times there will be need to withdraw **Xyce** releases from distribution. There are two general reasons that a release might be withdrawn.

First, old releases may become out of date as new releases become eligible for distribution.

Second, a release that is determined to contain significant issues may influence the CCB to decide to withdraw that specific release from distribution. In this case the release may be withdrawn immediately or it may be withdrawn at some time prior to the end the normal distribution cycle (i.e. after a patch release is available). In all cases release tracking information is referenced so that all customers of the release can be notified.

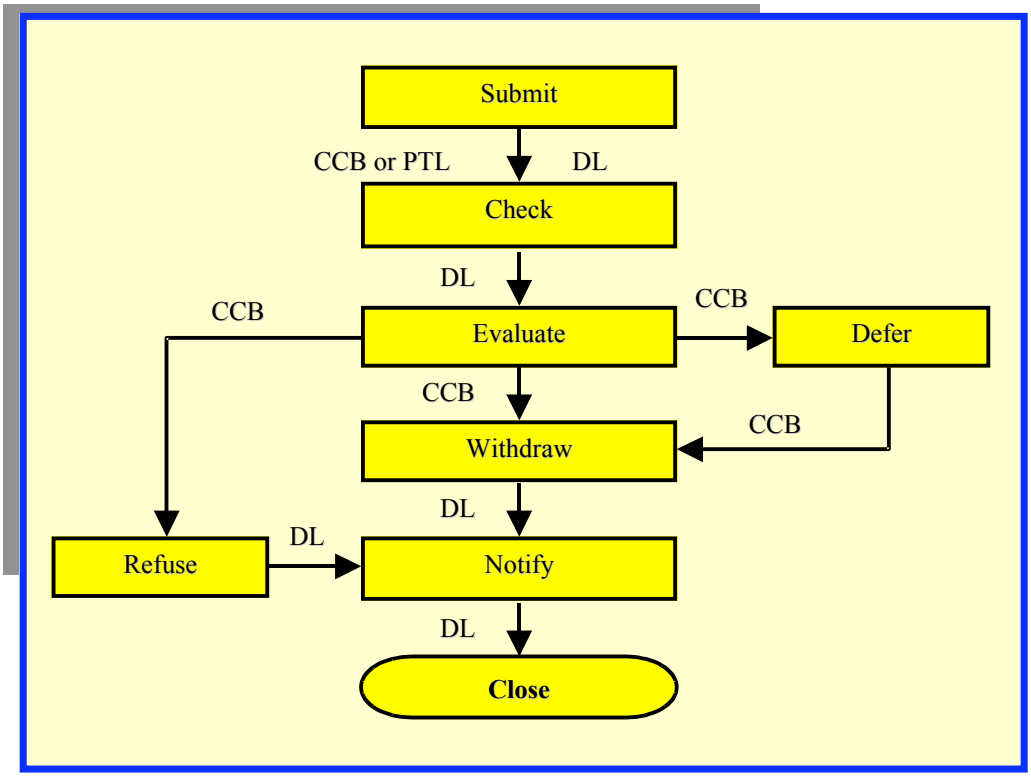

Figure 5. Illustration of the Withdraw process

Table 7 provides more detail on the withdraw activities, descriptions, and the role primarily responsible for the activity.

Activity	Description	Role Responsible
Submit	A formal request is made to withdraw a specific release from distribution The request must at a minimum specify the product, the release(s) impacted, and the reason for the request.	CCB or PTL
Check	The request is checked for completeness. The request may be returned to the originator or transitioned to the next activity.	DL
Evaluate	The evaluator(s) must determine if it is appropriate to withdraw the release. Requests may be refused, deferred, or approved.	CCB
Withdraw	The release is withdrawn from distribution.	DL
Notify	All interest parties are notified that the release has been withdrawn. Interested parties include: customers, PTL, and CCB. Customers must be advised that withdrawn releases are not supported.	DL
Close	Perform process improvement and administratively close the request	DL

Table 7. Withdrawing a Release

5. Roles

5.1 General

Roles represent a functional division of process responsibilities. Roles are defined based on the current activity, and the tasks that need to be accomplished.

Individuals may function in more than one role; however, an individual needs to know in what role he/she is functioning at any single point of time so they can verify that the proper set of tasks is being executed.

5.2 Role Definitions

Table 8 identifies the roles that are required for the Release Process and the Distribution Processes. Additional detail for each role is provided in the sections following Table 8.

Role	Description
Change Control Board (CCB)	A group of individuals who are responsible for making high-level decisions regarding releases.
Product Team Lead (PTL)	An individual who is responsible for the product architecture, product development, and managing resources assigned to product related activities.
Developer (DEV)	Responsible for designing, writing, and maintaining the source and related documentation files that comprise some part of a product.
QA	Individuals who are responsible for assuring that application software conforms to project standards, and for the testing and evaluating releases.
Release Lead (RL)	Responsible for overseeing and documenting all releases.
Distribution Lead (DL)	Responsibly for the majority of the tasks in the Distribution Process.

Table 8. Roles

5.3 Role Details

Table 8 defined the roles required for RDM. The following sections expand, clarify, and illustrate those roles.

CCB

The CCB has defacto management authority of overall project direction and development resources. The CCB is comprised (at a minimum) of a representative from management, a representative from product management, and the product team lead (PTL).

Examples of the tasks that fall under the authority of the CCB include:

- Approving and certifying release requests;
- Evaluating withdraw requests;
- Resolving project level technical issues;
- Managing project resource availability; and
- Approving release timing.

PTL

The product team lead is responsible for day-to-day oversight of product technical issues and resources assigned to the product.

Examples of the tasks that fall under the authority of the PTL include:

- Review and scope release requests;
- Plan releases; and
- Direct release development.

DEV

The developer is responsible for designing, writing, and documenting code.

Examples of the tasks that fall under the authority of the developer include:

- Receiving assignments from the PTL to create, enhance, or debug product functionality;

- DEVEL activity of the Release Development Lifecycle including identification of the elements that go into the STABLE baseline; and

- Notifying QA when the release is ready to be transitioned to the STABLE activity.

QA

Quality Assurance has several sub-roles. The QA Lead (QAL), QA Builder (QAB, and the QA Tester (QAT).

QAL

The QA Lead (QAL) is responsible for:

- Managing evaluation of the Release Development Lifecycle STABLE and QA activities; and

- Verifying that all work done to support a release relates to specific change requests.

QAB

The QA Builder (QAB) is responsible for creating QA product releases at the direction of the QAT. Activities include:

- Documenting which element versions are included in the STABLE and QA Release Development Lifecycle activities;

- Creating the STABLE and QA release baselines; and

- Performing and evaluating the STABLE and QA builds for targeted release environments.

QAT

The QA Tester (QAT) is responsible for preforming tasks to evaluate releases at the direction of the QAT. Activities include:

- Manual testing of releases; and

- Automated release testing.

RL

The release lead is responsible for overseeing the Release Process and for documenting all releases. Activities include:

- Ensuring inclusion of release notes and installation notes;
- Creating the PROD release baseline. Verifying contents of the PROD baseline;
- Documenting the date that the release becomes eligible for distribution; and
- Closing release requests.

DL

The distribution lead is responsible for the majority of the tasks during the Distribution Process. Activities include:

- Notifying of customers when a release becomes eligible for distribution;
- Notifying customers when a release is withdrawn from distribution; and
- Documenting and tracking distributions.

Customer

The customer has few (if any) official responsibilities related to RDM. Examples of the activities in which the customer may engage include:

- Obtaining appropriate licenses;
- Receiving and installing releases;
- Submitting product trouble reports;
- Submitting product enhancement requests; and
- Calling the help desk with issues.

6. Interfaces to Other Processes

Release and Distribution Management (RDM) has interfaces to other **Xyce** processes.

6.1 Issue Tracking

Release requests made be initiated as a result of defect reports or when the PTL requests a defect patch. Both require that an issue be reported to issue tracking. The QA Lead (QAL) may allow a release to proceed (with a defect) while reporting an issue (that will eventually generate another release request). The DL uses issue tracking to report media and packaging issues

The PTL (product team lead) is called the Team Lead (TL) in the Issue Tracking process (see [3]).

6.2 Requirements Management

Release requests may be initiated by a variety of sources. Release requests may include references to both new and existing requirements. Any new requirements that arise from release requests must be handled through the Requirements Management Process [2] .

All release request links to requirements need to be documented sufficiently to allow requirements base testing of release functionality.

The PTL (product team lead) is called the TL in the Requirements Management document [2] .

6.3 Third-Party Software (TPS)

The **Xyce** Parallel Electronic Simulator has dependencies on third-party software (TPS). The TPS required to build **Xyce** is typically utilized in the form of statically linked libraries that can be categorized in the following manner:

- Unmodified third party software (UTPS)
- Modified third party software (MTPS)
- **Xyce** specific external software (XSES)

Depending on which category of which a given library is a member, the maintenance and distribution of this software is handled slightly differently as is described in the Third-Party Software Configuration Management Plan [4] . Here we simply specify that TPS artifacts are baselined and tracked in coordination with the Release Process.

6.4 Configuration Management

Product release and product distribution are components of Configuration Management (CM). Proper implementation of RDM requires that another component of CM, version control, is properly defined and implemented. Since version control procedures are not defined in this document, the assumption is made that proper version control will be performed for all items that are needed for **Xyce** releases. That is, all instances of all controlled **Xyce** elements (e.g. source files, build files, test files, data files, third-party libraries, documentation, release notes,...) will be placed under version control.

The Release Management Process assumes that a Configuration Management Process is properly implemented. Version control of elements that compose releases is of particular importance to the Release Development Lifecycle DEVEL, STABLE, QA, and PROD activities.

References

[1] ASCI Apps Software Development Guide, May 2001.
[2] **Xyce** Requirements Management.
[3] **Xyce** Issue Tracking.
[4] **Xyce** Third Party Software Configuration Management Plan.

Appendix 1 Guidelines for Release Notes

Suggested Outline for Release Notes

The following is a generic outline for release notes for a software product. The philosophy implied by this generic outline is that the release notes will provide an overall description of the product and then provide information just about the current release of the product.

1.1 Scope/Product Definition

This section would have an overall description of the product. This description should be general enough that it applies to all releases of the product. In other words, this section is release-independent.

1.2 Hardware/Software Information

Supported Platforms

This section gives the currently supported hardware/operating-system combinations.

Hardware Requirements

Minimum hardware requirements required for running the code. These may be platform dependent.

Software Requirements

Minimum software requirements for running the code, including any standard or third party libraries required by the code.

1.3 Release Documentation

1.4 New Features and Enhancements

Highlights

This section describes major features of the product that should be highlighted. In essence this is the "sales pitch" section. Like the Product Description section, the highlights described in this section should be general enough to apply to all releases of the product. Features specific to a release that should be highlighted will be included in the next section, New Features.

Package-Specific Features and Enhancements

This section describes the new features and enhancements at the UML package level.

1.5 Defects Fixed in this Release

This section details the defects that were fixed specifically in the noted release. Since a sub release should include all the fixes in the patches that preceded the subareas, descriptions of the defects fixed by those patches do not have to be included.

1.6 Known Defects and Workarounds

This section is optional. It highlights significant defects that were not fixed in the release. This section should be included if there are some major defects that are still outstanding or if before the completion of the release, a release was advertised to fix certain defects, but the fixes were not included. It should also include workarounds, if available.

Appendix 2 Release Certification Tests

Checklist for Release Certification

The following checklist outlines the test necessary for certification of a **Xyce** release. As the **Xyce** code matures, this list will expand and evolve to match its capabilities. These tests encompass the standard **Xyce** test suite and add in several other tests which exercise other aspects of **Xyce** not covered by the current regression tests.

Test Circuit or Suite	Platform / Mode	Description	Owner	Completion Date
Regression Test Suite	Supported platforms	Standard **Xyce** nightly regression test suite	Tom Russo	
Regression Test Suite	Supported platforms / Debug mode	Standard **Xyce** nightly regression test suite run in Debug mode	Tom Russo	
Bug Tests	Support platforms	Test cases which demonstrate bug fixes for this release	The "owner" of the bug to which the test case is tied. This person is tracked in the **Xyce** bugzilla tracking tool	
spice_ssu	Supported platforms	Bill Ballard's demo circuit	Tom Russo	
RHP Adder	Supported parallel platforms	Circuit run in parallel on practical ranges of numbers of processors	Rob Hoekstra	
RHP Multiplier	Supported parallel platforms	Circuit run in parallel on practical ranges of numbers of processors	Rob Hoekstra	
FPGA2	Supported parallel platforms	Circuit run in parallel on practical ranges of numbers of processors	Rob Hoekstra	
Micro-strip Transmission Line	Supported parallel platforms	Circuit run in parallel on practical ranges of numbers of processors	Rob Hoekstra	

Release Certified (project leader): ☐ Release certified

Certification Date: _____

General Directions:

The release certification date should be filled in immediately to initiate this process. The initiation date must be the date of printing. The indicated steps may be completed in any reasonable order. The process shall be considered complete when all steps have been completed.

Specific Directions:

No specific form of review is required (i.e. Fagan Inspection). The project leader shall determine the scope of the planning and the method of conducting the review. The project leader shall notify planning participants and provide them with the applicable material.

This checklist is complete when the project leader determines its disposition. The project leader may consult with the participants to determine the final disposition.

QA

Process Ownership:

This document and associated procedure are owned by the Project Leader who must approve any changes. This document is version 1.0 and supercedes all previous versions. This document is under version control. All previous versions will print with a banner "OBSOLETE".

Authority to perform process:

Any team member may initiate a meeting using this procedure.

If this process or associated document is superceded after an activity has been initiated, the process shall be completed under this version unless specifically notified otherwise by the team leader.

Stage Entrance and Stage Exit:

The stage entrance is the process initiation by any authorized person. All necessary process inputs are built into the checklist. The stage exit is the completion of this form (all steps completed and metrics entered.) Note (see Specific Directions above) that the project leader can terminate the process once the disposition can be determined to avoid wasted effort.

Appendix 3 Release Process Checklists

Checklists for Release Process Activities

The following are a set of checklist tables for use with **Xyce** releases to help ensure completion and documentation of necessary activities. Following each table is a set of directions as well as QA information for the list. Note that these checklist are available as separate documents to be used and archived for each release cycle.

3.1 Release Planning Checklist

Release Number: _____

Release Planning Date: _____

Planning Participants: _____

Activity	Description	Owner	Completion Date
Plan Target Capabilities	Generate list of target capabilities for current release. If possible, these should be traced to specific requirements and/or change requests.		
Plan Target Functionality	Generate list of target functionality for current release. If possible, these should be traced to specific requirements and/or change requests.		
Develop Acceptance Criteria	Specify acceptance criteria for each target capability and/or functionality.		
Activity Dates	Establish dates for each activity: ▪ Code repository branch ▪ Production build ▪ Release notes completion ▪ Release certification ▪ Release notification Note that these activities may be specified in a relative manner (e.g., relative to the process initiation or an activity completion).		

Plan Completion (project leader): ☐ Release plan complete

Completion Date: _____

General Directions:

The release planning date and participants should be filled in immediately to initiate this process. The initiation date must be the date of printing. The

indicated steps may be completed in any reasonable order. The process shall be considered complete when all steps have been completed.

Specific Directions:

No specific form of review is required (i.e. Fagan Inspection). The project leader shall determine the scope of the planning and the method of conducting the review. The project leader shall notify planning participants and provide them with the applicable material.

This checklist is complete when the project leader determines its disposition. The project leader may consult with the participants to determine the final disposition.

QA

Process Ownership:

This document and associated procedure are owned by the Project Leader who must approve any changes. This document is version 1.0 and supercedes all previous versions. This document is under version control. All previous versions will print with a banner "OBSOLETE".

Authority to perform process:

Any team member may initiate a meeting using this procedure.

If this process or associated document is superceded after an activity has been initiated, the process shall be completed under this version unless specifically notified otherwise by the team leader.

Stage Entrance and Stage Exit:

The stage entrance is the process initiation by any authorized person. All necessary process inputs are built into the checklist. The stage exit is the completion of this form (all steps completed and metrics entered.) Note (see Specific Directions above) that the project leader can terminate the process once the disposition can be determined to avoid wasted effort.

3.2 Release Configuration Management (RCM) Checklist

Release Number: _____

RCM Initiation Date: _____

RCM Initiation Owner: _____

Activity	Description	Owner	Completion Date
Baseline CM Code Repository	The code CM repository should be baselined (tagged) according to the labeling convention established in the Release and Distribution Management.		
Create Release Branch	The code CM repository shall be branched in support of the Release Development Lifecycle.		
Track Release Development Lifecycle	The RDL shall be tracked in a manner consistent with the Release and Distribution Management document. This shall be accomplished by baselining each promotion through the RDL.		
Baseline Final Release Configuration	At the completion of the RDL (PROD stage), the repository shall be tagged according to the labeling convention established in the Release and Distribution Management guide.		

RCM Completion (project leader): ☐ RCM complete

Completion Date: _____

General Directions:

As release activities proceed through the RDL, this checklist must be updated/completed. The initiation date and owner shall be filled in immediately to initiate this process. The initiation date must be the date of printing. The indicated steps may be completed in any reasonable order. The process shall be considered complete when all steps have been completed.

QA

Process Ownership:

This document and associated procedure are owned by the Project Leader who must approve any changes. This document is version 1.0 and supercedes all previous versions. This document is under version control. All previous versions will print with a banner "OBSOLETE".

Authority to perform process:

Any team member may initiate a meeting using this procedure.

If this process or associated document is superceded after an activity has been initiated, the process shall be completed under this version unless specifically notified otherwise by the team leader.

Stage Entrance and Stage Exit:

The stage entrance is the process initiation by any authorized person. All necessary process inputs are built into the checklist. The stage exit is the completion of this form (all steps completed and metrics entered.) Note (see Specific Directions above) that the project leader can terminate the process once the disposition can be determined to avoid wasted effort.

3.3 Release Development Lifecycle (RDL) Checklist

Release Number: _____

RDL Initiation Date: _____

RDL Initiation Owner: _____

Activity	Description	Owner	Completion Date
DEVEL	During DEVEL for a given RDL, work is performed to add, enhance, fix or remove functionality. Completion of this stage is determined by the assigned developers and results in promotion to the STABLE stage.		
STABLE	Releases that have reached this activity are ready for element and component level testing. This is performed according to associated test plans. Failure in this stage requires returning to the DEVEL stage with associated change requests will be entered.		
QA	During this phase, acceptance tests are performed on all target platforms. In addition, testing and related activities may be performed by other QA individuals. Failure here will result in returning to the STABLE stage and possibly the DEVEL stage with associated change requests will be entered.		
PROD	At this final stage, the product is ready for certification. Here, release notes and other documentation may be initiated.		

RDL Completion (project leader): ☐ RDL complete

Completion Date: _____

General Directions:

For each of possibly multiple passes (minimum one) through the RDL, this checklist must be completed. The initiation date and owner shall be filled in

immediately to initiate this process. The initiation date must be the date of printing. The indicated steps may be completed in any reasonable order. The process shall be considered complete when all steps have been completed.

QA

Process Ownership:

This document and associated procedure are owned by the Project Leader who must approve any changes. This document is version 1.0 and supercedes all previous versions. This document is under version control. All previous versions will print with a banner "OBSOLETE".

Authority to perform process:

Any team member may initiate a meeting using this procedure.

If this process or associated document is superceded after an activity has been initiated, the process shall be completed under this version unless specifically notified otherwise by the team leader.

Stage Entrance and Stage Exit:

The stage entrance is the process initiation by any authorized person. All necessary process inputs are built into the checklist. The stage exit is the completion of this form (all steps completed and metrics entered.) Note (see Specific Directions above) that the project leader can terminate the process once the disposition can be determined to avoid wasted effort.

3.4 Release Certification Checklist

Release Number: _____

Certification Initiation Date: _____

Certification Participants: _____

Activity	Description	Owner	Completion Date
Perform Acceptance Tests	Acceptance test, as specified in the Release Plan, are performed and documented.		
Complete and Review Documentation	Documentation (Release Notes, updated Guides, etc.) is generated and reviewed by assigned participants.		
Certify Release	Release is certified assigned reviewers and management. The review form is archived.		

Certification Completion (project leader): ☐ Release certification complete

Completion Date: _____

Certification Signatures:

QAL: _____

PTL: _____

PI: _____

Mgt. Rep.: _____

General Directions:

The certification date and participants should be filled in immediately to initiate this process. The initiation date must be the date of printing. The indicated steps may be completed in any reasonable order. The process shall be considered complete when all steps have been completed.

Specific Directions:

No specific form of review is required (i.e. Fagan Inspection). The project leader shall determine the scope of the planning and the method of conducting

the review. The project leader shall notify planning participants and provide them with the applicable material.

This checklist is complete when the project leader determines its disposition. The project leader may consult with the participants to determine the final disposition.

QA

Process Ownership:

This document and associated procedure are owned by the Project Leader who must approve any changes. This document is version 1.0 and supercedes all previous versions. This document is under version control. All previous versions will print with a banner "OBSOLETE".

Authority to perform process:

Any team member may initiate a meeting using this procedure.

If this process or associated document is superceded after an activity has been initiated, the process shall be completed under this version unless specifically notified otherwise by the team leader.

Stage Entrance and Stage Exit:

The stage entrance is the process initiation by any authorized person. All necessary process inputs are built into the checklist. The stage exit is the completion of this form (all steps completed and metrics entered.) Note (see Specific Directions above) that the project leader can terminate the process once the disposition can be determined to avoid wasted effort.

3.5 Distribution Management Checklist

Release Number: _____

Distribution Initiation Date: _____

Distribution Owner: _____

Activity	Description	Owner	Completion Date
Build Platform Executables	For each supported platform, build executables from the baselined release.		
Distribute Release	Distribute supported executables and associated artifacts according to distribution plan.		
Notify Customers	Notify customers of Release and provide release artifacts (either actual artifacts or instructions on access).		

Distribution Completion (project leader): ☐ Distribution complete

Completion Date: _____

General Directions:

Complete this checklist as part of the overall distribution process. The initiation date and owner shall be filled in immediately to initiate this process. The initiation date must be the date of printing. The indicated steps may be completed in any reasonable order. The process shall be considered complete when all steps have been completed.

QA

Process Ownership:

This document and associated procedure are owned by the Project Leader who must approve any changes. This document is version 1.0 and supercedes all previous versions. This document is under version control. All previous versions will print with a banner "OBSOLETE".

Authority to perform process:

Any team member may initiate a meeting using this procedure.

If this process or associated document is superceded after an activity has been initiated, the process shall be completed under this version unless specifically notified otherwise by the team leader.

Stage Entrance and Stage Exit:

The stage entrance is the process initiation by any authorized person. All necessary process inputs are built into the checklist. The stage exit is the completion of this form (all steps completed and metrics entered.) Note (see Specific Directions above) that the project leader can terminate the process once the disposition can be determined to avoid wasted effort.

Distribution

1 MS 0318
Paul Yarrington, 09230

1 MS 0316
Sudip Dosanjh, 09233

1 MS 0525
Paul V. Plunkett, 01734

1 MS 0835
J. Michael McGlaun, 09140

1 MS 0828
Martin Pilch, 09133

1 MS 0139
Stephen E. Lott, 09905

5 MS 0316
Scott A. Hutchinson, 09233

1 MS 0316
Eric R. Keiter, 09233

1 MS 0316
Robert J. Hoekstra, 09233

1 MS 0316
Joseph P. Castro, 09233

1 MS 0316
Roger Pawlowski, 09233

1 MS 0316
Richard Schiek, 09233

1 MS 1111
John N. Shadid, 09233

1 MS 0847
Tim Trucano, 09211

1 MS 0847
Bart van Bloemen Waanders, 09211

1 MS 0196
Elebeoba May, 09212

1 MS 1110
Todd Coffey, 09214

1 MS 1110
David Day, 09214

1 MS 1110
Mike Heroux, 09214

1 MS 1110
James Willenbring, 09214

1 MS 1111
Karen Devine, 09215

1 MS 0525
Thomas V. Russo, 01734

1 MS 0525
Lon Waters, 01734

1 MS 0525
Carolyn Bogdan, 01734

5 MS 1138
C. Mike Williamson, 06536

1 MS 1138
Rebecca Arnold, 06536

1 MS 1138
Smitha Sam, 06536

1 MS 1138
Ann Hodges, 06536

1 MS 1138
Gary Froelich, 06536

1 MS 1138
Michael Tebo, 06536

1 MS 0525
Ronald Sikorksi, 01734

1 MS 0660
Roger F. Billau, 09519

1 MS 1081
Charles E. Hembree, 01739

1 MS 0807
Davod N. Shirley, 09328

1 MS 9409
William P. Ballard, 08730

1 MS 9202
Kathryn R. Hughes, 08205

1 MS 9202
Kenneth D. Marx, 08205

1 MS 9202
Stephen L. Brandon, 08205

1 MS 1179
Leonard Lorence, 15341

1 MS 9018
Central Technical Files,
8945-1

2 MS 0899
Technical Library, 9616